



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

EVERTON MURILO BERNARDO DA SILVA

**SISTEMA PARA GESTÃO DE ARQUIVOS UTILIZANDO O
FRAMEWORK SPRING SECURITY**

ASSIS - SP

2013

EVERTON MURILO BERNARDO DA SILVA

**SISTEMA PARA GESTÃO DE ARQUIVOS UTILIZANDO O
FRAMEWORK SPRING SECURITY**

Projeto de Desenvolvimento de software apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientado: Everton Murilo Bernardo da Silva

Orientador: Me. Douglas Sanches da Cunha

ASSIS - SP

2013

FICHA CATALOGRÁFICA

SILVA, Everton Murilo Bernardo da

Sistema para Gestão de Arquivos utilizando o Framework Spring Security /
Everton Murilo Bernardo da Silva. Fundação Educacional do Município de Assis –
FEMA – Assis, 2013.

36p.

Orientador: . Me. Douglas Sanches Cunha.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis
– IMESA.

1. Software. 2. Web. 3. Framework. 4. JAVA. 5. Armazenamento.

CDD: 001.61
Biblioteca da FEMA

SISTEMA PARA GESTÃO DE ARQUIVOS UTILIZANDO O FRAMEWORK SPRING SECURITY

EVERTON MURILO BERNARDO DA SILVA

Projeto de Desenvolvimento de software apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientador: Me. Douglas Sanches da Cunha

Analisador: Célio Desiro

ASSIS - SP

2013

DEDICATÓRIA

Dedico à minha família que me incentivou a realizar o projeto dando total apoio. Aos meus amigos que me ajudaram nas dúvidas e à turma que me proporcionou momentos únicos durante o curso sempre com muita alegria.

AGRADECIMENTOS

Agradeço primeiramente a Deus por permitir realizar o curso e concluir essa longa jornada.

Ao meu orientador Me. Douglas Sanches Cunha por me instruir a fazer o meu melhor.

Aos outros professores pelo conhecimento transmitido.

Aos meus familiares e amigos que me ajudaram em todos os momentos com muito incentivo e compreensão, sempre me motivando a me dedicar ao trabalho.

RESUMO

Este trabalho descreve um sistema de armazenamento e gerenciamento de arquivos em nuvem, tendo como objetivo facilitar o uso e armazenamento de arquivos através da web, proporcionando assim uma maior agilidade e facilidade no manuseio de dados e informações. Este sistema foi desenvolvido com o recurso de um framework muito importante, o *Spring Security* que fará toda a segurança do sistema. A linguagem de programação usada para o desenvolvimento da aplicação é a tecnologia JAVA, e com todo o conhecimento que por mim foi adquirido nesses três anos de graduação cursados nesta faculdade.

Palavras-chave: Software; Web; Framework; JAVA; Armazenamento.

ABSTRACT

This work describes a system for storage and management files in cloud, the objective is facilitate the use and storage of files through the web, providing greater agility and facility in data handling and information. This system was developed with the use of a very important framework, the Spring Security that will make all the system security. The programming language used for the application development is JAVA technology, and with all the knowledge that has been acquired by me in these three years of graduation realized at this college.

Keywords: Software, Web, Framework, JAVA; Storage.

LISTA DE ILUSTRAÇÕES

Figura 1 - Configuração de acesso ao banco de dados	16
Figura 2 – Configuração de autenticação e interceptação	17
Figura 3 – Estrutura do desenvolvimento	17
Figura 4 – Autenticação de usuário	18
Figura 5 – Web.xml	19
Figura 6 – Importação de bibliotecas do IDE	20
Figura 7 – Importação de JAR's	21
Figura 8 – Cronograma de Desenvolvimento	30
Figura 9 – Diagrama de Caso de Uso	31
Figura 10 – Caso de Uso Manter Arquivo	32
Figura 11 – Diagrama de sequencia – Manter Arquivo	33
Figura 12 – Caso de Uso Manter Usuário	34
Figura 13 – Diagrama de sequencia – Manter usuário	36
Figura 14 – Caso de Uso Manter Arquivo - Administrador	37
Figura 15 – Diagrama de sequencia – Upload de arquivo	39
Figura 16 – Diagrama de sequencia – Download de arquivo	40
Figura 17 – Diagrama de Classes	41
Figura 18 – Modelo de Entidade-Relacionamento	42
Figura 19 – Processo de Upload	43
Figura 20 – Processo de Download	43

SUMÁRIO

1. INTRODUÇÃO	12
1.1. OBJETIVO	13
1.2. PÚBLICO ALVO	13
1.3. JUSTIFICATIVA	14
1.4. MÉTODO DE DESENVOLVIMENTO	14
1.5. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA	15
2. SPRING FRAMEWORK	16
2.1. APPLICATION CONTEXT	16
2.2. WEB-XML	18
2.3. BIBLIOTECAS	19
3. ESPECIFICAÇÃO	22
3.1. O PROJETO	22
3.2.1. JAVA	23
3.2.2. NetBeans 7.2.1	24
3.2.3. Spring Security	25
3.2.4. JavaServer Faces	25
3.2.5. MySQL	25
4. ANÁLISE DE REQUISITOS	27
4.1. REQUISITOS FUNCIONAIS	27
4.2. REQUISITOS NÃO FUNCIONAIS	28
4.2.1. Requisito de Segurança	28
4.2.2. Requisito de Desempenho	28
4.2.3. Requisito de Compatibilidade	28
4.2.4. Requisito de Implementação	29

4.3.	ORÇAMENTO DO PROJETO.....	29
□	Equipamentos	29
□	Softwares	29
4.4.	CRONOGRAMA DO PROJETO	30
4.5.	DIAGRAMAS.....	31
4.5.1.	DIAGRAMA DE CASO DE USO MANTER ARQUIVO	32
4.5.2.	DIAGRAMA DE SEQUÊNCIA MANTER ARQUIVO	33
4.5.3.	DIAGRAMA DE CASO DE USO MANTER USUÁRIO	34
4.5.4.	DIAGRAMA DE SEQUÊNCIA MANTER USUÁRIO	36
4.5.5.	DIAGRAMA DE CASO DE USO.....	37
4.5.6.	DIAGRAMA DE SEQUÊNCIA UPLOAD ARQUIVO	39
4.5.7.	DIAGRAMA DE SEQUÊNCIA DOWNLOAD ARQUIVO	40
4.5.8.	DIAGRAMA DE CLASSE	41
4.5.9.	MODELO DE ENTIDADE-RELACIONAMENTO	42
4.5.10.	PROCESSOS PRINCIPAIS	43
5.	CONCLUSÃO.....	44
	REFERÊNCIAS.....	45

1. INTRODUÇÃO

O sistema será desenvolvido para web objetivando criar um software de gerenciamento e armazenamento de arquivos na nuvem, para não ser necessário utilizar mídias físicas. A principal característica do sistema será aplicada através de um *framework* que trará esse segurança para os usuários do sistema.

Nos dias de hoje as pessoas utilizam muito mídias físicas como *CD's*, *DVD's*, e *pendrive's* para armazenar, gerenciar e transportar seus arquivos para outros dispositivos. O sistema tentará aprimorar esse conceito levando em consideração principal o uso da computação em nuvem para o armazenamento dos dados e a facilidade de transportar informações de um lugar para o outro utilizando apenas a Internet.

A Internet torna-se um pouco perigosa devido a grande quantidade de *hackers* e vírus navegando na rede. Visando solucionar esse problema, o sistema contará com uma ferramenta de segurança para proteger a conta do usuário, proporcionando segurança e confiabilidade na aplicação (*software*). Com isso, o usuário cadastrado no sistema terá a opção de entrar em qualquer dispositivo dotado de acesso à Internet e ter acesso a todos os seus dados com total segurança.

O *Spring Security* ajudará a melhorar a segurança e nas questões de limitações que podem ser impostas ao sistema, como habilitar e desabilitar funções para usuários.

1.1. OBJETIVO

O sistema tem como objetivo inicial armazenar arquivos em nuvem, mas o foco principal do sistema é a segurança, prover tal segurança aos arquivos armazenados em cada conta de usuário, dando-lhe a privacidade necessária e reafirmando a confiabilidade empregada no sistema de modo que visa oferecer os seus serviços de modo rápido e prático. Com os arquivos armazenados em nuvem os usuários poderão economizar espaços em periféricos, diminuindo o risco de perda de arquivos.

A Internet de modo geral facilita muito a vida das pessoas, e o surgimento da mobilidade torna ainda mais possível a implementação do sistema, que pode ser acessado via web e o usuário pode fazer *download* de seus arquivos em qualquer mídia ou dispositivo, facilita o transporte dessa informação sem custo algum.

1.2. PÚBLICO ALVO

O público utilizará um sistema genérico, pois se pretende atingir desde usuários que desejam armazenar seus arquivos para desocupar espaço nos meios de armazenamento físico até usuários que pretendem usá-lo para fins de trabalho ou mesmo de compartilhamento. Os usuários terão liberdade de escolher se desejam ou não compartilhar seus arquivos aos demais usuários, dando assim opção de escolha e privacidade na conta do usuário.

1.3. JUSTIFICATIVA

A principal necessidade do sistema é impor a segurança de modo geral, onde o *framework* a ser aplicado trará não só segurança para o próprio sistema como os dados dos usuários nele armazenados, protegendo tanto contra ataques de hackers quanto aos vírus existentes na internet. E pode-se definir a partir do *Spring Security* permissões de usabilidade do sistema, onde é possível liberar acessos ou não a determinados recursos oferecidos pelo sistema.

1.4. MÉTODO DE DESENVOLVIMENTO

O sistema será desenvolvido em plataforma Java para web utilizando o IDE Netbeans 7.1 para a criação do projeto. O foco principal será no *framework* de segurança a ser utilizado, o *Spring Security*, voltado para a proteção do sistema e segurança dos dados armazenados. Além disso, o sistema contará com um banco de dados *MySQL Server* para armazenamento de seus dados e o servidor web será o *TOMCAT*.

Para a preparação da análise do projeto, o *Astah* será a ferramenta. Para a formatação e edição da documentação do sistema e do trabalho de modo geral, será utilizado o Word.

1.5. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA

O sistema contará com o método de desenvolvimento orientado a objeto, que facilitará a organização dos códigos dentro da aplicação.

- Técnica para o levantamento de requisitos

A técnica de apoio escolhida para auxiliar na elicitação dos requisitos foi a oportunidade de desenvolver um software capaz de auxiliar as pessoas. Deste modo, podem-se observar as características que o software deverá ter para a aceitação de mercado, os benefícios esperados, os desejos, as restrições e funcionalidades.

- Especificação dos requisitos.

Nesta etapa, os requisitos obtidos anteriormente são descritos e especificados com nível de detalhamento adequado ao sistema.

- Validação dos requisitos.

Após a especificação dos requisitos, a etapa de validação consiste em verificar se os requisitos obtidos estão de acordo com os propostos no começo do processo.

Se tudo estiver coerente, a validação é realizada e o software entra em processo de desenvolvimento.

2. SPRING FRAMEWORK

O *Spring Framework* é uma ferramenta que auxilia no desenvolvimento de softwares para plataformas web baseadas e linguagem *Java*, este *framework* possui classes, bibliotecas e métodos que permitem ao desenvolvedor buscar funções específicas sobre ele sem precisar de programação, o software oferece isto para ele. O *Spring* é baseado na inversão de controle e injeção de dependências dentro da aplicação.

2.1. APPLICATION CONTEXT

Para que o *Spring* seja implementado e configurado corretamente para funcionar na aplicação devemos tomar alguns cuidados referente a isso. Primeiramente começamos criando um arquivo *XML* dentro da pasta *WEB-INF* da sua aplicação, daremos o nome a esse arquivo como *applicationContext*. Dentro deste arquivo é preciso configurar três partes principais para que a autenticação e a interceptação funcionem. A primeira parte é configurada a interação com o banco de dados da aplicação, onde será colocado o nome do banco, usuário e senha (WEISSMANN, 2013).

```
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName" value="com.mysql.jdbc.Driver" />
  <property name="url" value="jdbc:mysql://localhost/sgaspring" />
  <property name="username" value="root" />
  <property name="password" value="murilo" />
</bean>
```

Figura 1 – Configuração de acesso ao banco de dados

Depois de feito essa configuração é preciso então definir a interceptação das páginas de acordo com a permissão de cada usuário definido no cadastro. Essa autenticação será por usuário senha, buscando sempre o campo de permissão para validar o acesso do usuário ao sistema (WEISSMANN, 2013).

```
<sec:http auto-config="true" access-denied-page="/negado.xhtml" >
<sec:form-login login-page="/login.xhtml" authentication-failure-url="/login.xhtml?erro=true" />
<sec:intercept-url pattern="/admin/**" access="ROLE_Administrador" />
<sec:intercept-url pattern="/usuario/**" access="ROLE_Usuario,ROLE_Administrador" />
<sec:intercept-url pattern="/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
```

Figura 2 – Configuração de autenticação e interceptação

Essa primeira parte está totalmente voltada para a segurança da aplicação, isso significa que se a autenticação não for liberada para o usuário nessa parte qualquer maneira que tente para acessar o sistema será negada, pois o *framework* não reconhece uma autenticação válida para dar continuidade de acesso ao sistema.

A segurança do sistema é feita justamente nessa parte, os as *TAG's* "sec:" definem as opções de segurança que o software terá. A interceptação de páginas é realizada na terceira linha do código onde é definido por uma estrutura de pastas onde contém as *URL's* válidas, ou seja, as pastas definem o perfil e as páginas que o usuário do perfil pode acessar (WEISSMANN, 2013).

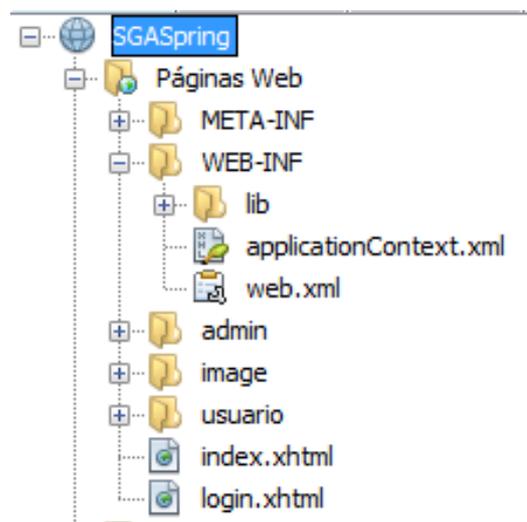


Figura 3 – Estrutura do desenvolvimento

Para terminar essa primeira parte da interceptação de usuários e autenticação das páginas no *applicationContext.xml* será feito o último código de segurança da aplicação que será onde selecionara o usuário, senha e permissão no banco de dados e será liberado ou negado o acesso as *URL*'s configuradas por permissão na segunda parte do código do *applicationContext.xml* (WEISSMANN, 2013).

```
<sec:authentication-manager>
  <sec:authentication-provider>
    <sec:jdbc-user-service data-source-ref="dataSource"
      users-by-username-query="SELECT login, senha, 'true' as enable FROM usuario WHERE login=?"
      authorities-by-username-query="SELECT login, concat('ROLE_',permissoao) FROM usuario WHERE login=?" />
  </sec:authentication-provider>
</sec:authentication-manager>
```

Figura 4 – Autenticação de usuário

Quando o usuário tenta o acesso ao sistema através da tela de login, essa parte do código é a responsável por realizar essa autenticação, comparando login e senha que o usuário colocou com os dados cadastrados no banco, ou seja, se houver o usuário cadastro o banco a autenticação será realizada e passa-se para a segunda parte do código onde será realizada a parte de interceptação de páginas de acordo com a permissão do usuário autenticado no sistema (WEISSMANN, 2013).

2.2. WEB-XML

O *web.xml* é outra parte da configuração do *Spring* muito importante, pois é aqui que são definidos as propriedades de referências do *framework*, ou seja, referências do *framework* para que funcione corretamente todas as configurações realizadas no *applicationContext.xml* (WEISSMANN, 2013).

```

<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>

<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>FORWARD</dispatcher>
</filter-mapping>

<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/applicationContext.xml
  </param-value>
</context-param>

<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

```

Figura 5 – Web.xml

2.3. BIBLIOTECAS

As bibliotecas referentes ao funcionamento do *Spring Framework* podem ser encontradas pelo desenvolvedor através do IDE que esteja utilizando, que no caso o *NetBeans* possui uma biblioteca com os *JAR's* que devem ser incluídos na sua aplicação para que as referencias colocas no web.xml e as configurações de autenticação e interceptação sejam reconhecidas e realizadas (WEISSMANN, 2013).

Java Archive (JAR) é um arquivo compactado usado para distribuir um conjunto de classes Java, um aplicativo Java, ou outros itens como imagens, XML's, entre outros. Disponível em: <<http://docs.spring.io/spring/docs/4.0.0.BUILD-SNAPSHOT/spring-framework-reference/htmlsingle/>>. Acessado em: 03 ago, 2013.

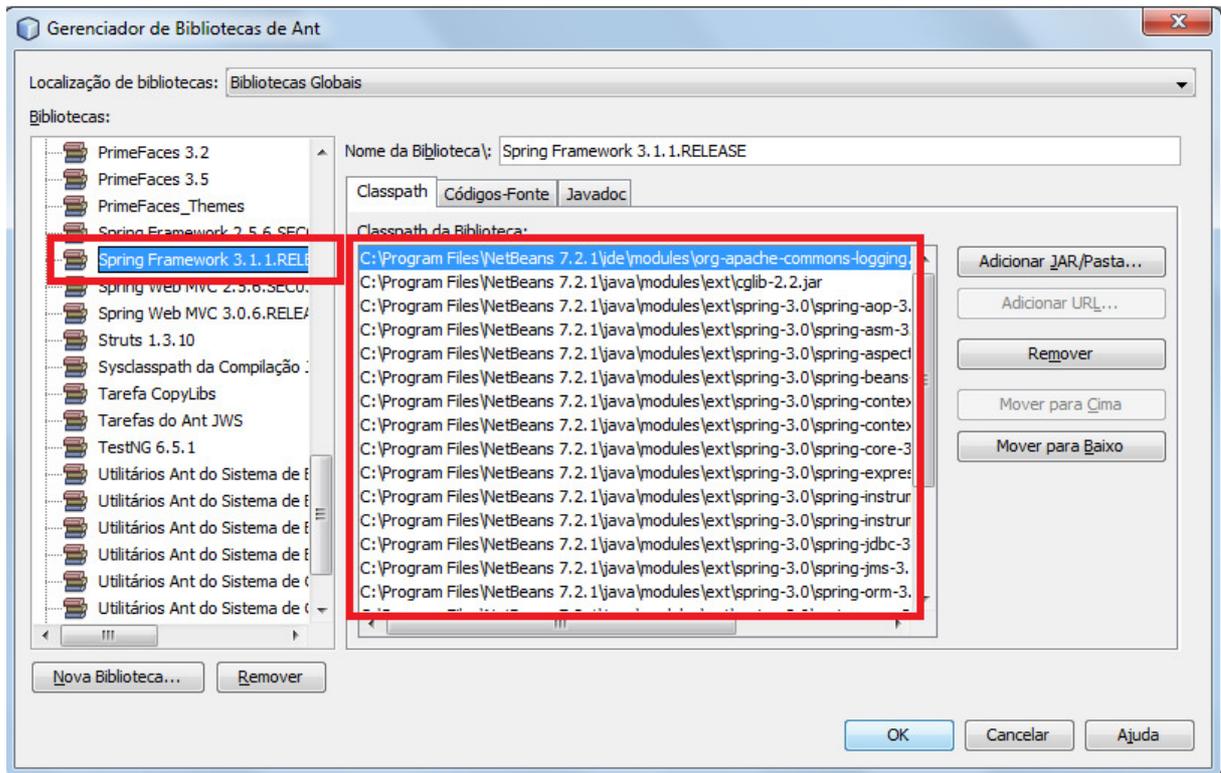


Figura 6 – Importação de bibliotecas do IDE

Mas se não conter essa biblioteca no IDE também pode ser incluído na aplicação os *JAR's*. Pode realizar o download destes arquivos pela página do *Spring Framework* onde pode ser incluído posteriormente juntamente com as bibliotecas do sistema. Todos os *JAR's* devem ser importados corretamente, pois se por ventura faltar alguns destes a aplicação poderá apresentar falhas e não funcionara corretamente. Disponível em: <<http://docs.spring.io/spring/docs/4.0.0.BUILD-SNAPSHOT/spring-framework-reference/htmlsingle/>>. Acessado em: 03 ago, 2013.

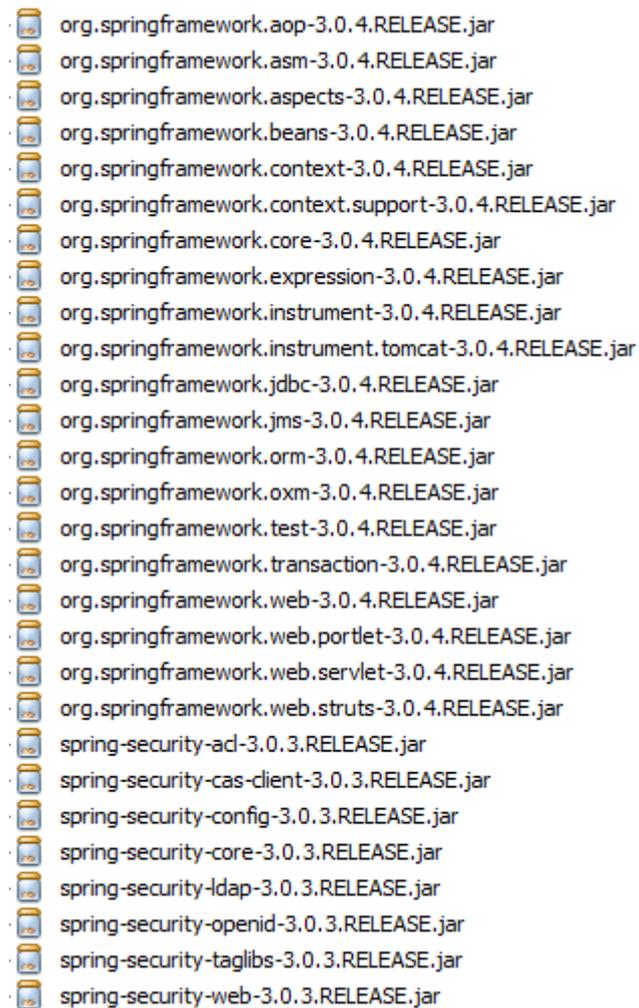


Figura 7 – Importação de JAR's

3. ESPECIFICAÇÃO

Este capítulo traz a especificação principal e funcional do software, onde será mostrado o principal objetivo e função que o *framework* oferece para o programador no desenvolvimento da aplicação.

3.1. O PROJETO

O sistema consiste em um software voltado para a plataforma Web que possibilitará ao administrador gerenciar arquivos armazenados por outros usuários do sistema, podendo se assim achar conveniente mantê-los ou excluí-los, e os usuários poderão armazenar os seus arquivos em um servidor em nuvem, gerenciando a sua própria conta. O *Framework Spring Security* trará maior segurança ao sistema, delimitando assim as ações do usuário, não permitindo que o mesmo acesse recursos indevidos, ou mesmo que algum usuário do sistema venha a invadi-lo e se apoderar de algum arquivo. O *JavaServer Faces* ajudará no desenvolvimento de páginas mais dinâmicas e rápidas, deixando a interface o mais compreensível possível para o usuário.

3.2. FERRAMENTAS DE DESENVOLVIMENTO

As metodologias usadas são contemporâneas e facilitam o uso do sistema por parte do usuário, pois possui uma interface mais limpa, o sistema em si fica mais rápido, pois o sistema irá rodar em plataforma web e segurança que é o principal.

3.2.1. JAVA

Java é uma linguagem computacional completa, adequada para o desenvolvimento de aplicações baseadas na rede Internet, redes fechadas ou ainda programas *stand-alone*.

A linguagem Java foi desenvolvida na década de noventa pela empresa *Sun Microsystems*, por uma equipe de programadores liderada por James Gosling. Essa linguagem é orientada a objetos e compilada por uma máquina virtual onde é executada em *bytecode* (GOMES, 2008).

O Java esta ganhando destaque na atualidade devido à plataforma de desenvolvimento para web, onde permite a migração de softwares desenvolvidos para desktop's e passa para a web, onde os softwares não ocupam espaço na memória da máquina. Dentre as principais vantagens é a portabilidade, que proporciona que o software seja desenvolvido em um computador com um determinado software, mas, funciona em qualquer sistema operacional e em qualquer *browser*. Como todo software desenvolvido tem que ter restrições e comunicações, o Java disponibiliza bibliotecas que permitem e facilitam a cooperação em alguns protocolos de rede e transmissão de dados como o TCP/IP, FTP e HTTP; a segurança também está presente nessa plataforma, podendo

executar restrições no software dependendo da estruturação do software (DEITEL, 2010).

Pode-se também ser desenvolvidos programas multitarefas e distribuídos, onde permite varias linhas de execução em um mesmo software; possui uma sintaxe parecida com as linguagens C/C++, e a sua principal característica (que praticamente define a orientação em objetos) a carga dinâmica de código, onde o programa é composto por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento da utilização (DEITEL, 2010).

3.2.2. NetBeans 7.2.1

O *NetBeans* foi criado em 1996, pela empresa Sun Microsystems. Consiste em um ambiente de desenvolvimento multiplataforma, ou seja, uma ferramenta que auxilia os programadores a desenvolver softwares, instalar aplicações, compilar os projetos de forma estruturada para simplificar o desenvolvimento e aumentar a produtividade, diminuindo os erros de programação (GOMES, 2008).

Atualmente o *NetBeans* é uma ferramenta de desenvolvimento gratuita, disponível para download em vários idiomas, fornecendo uma base sólida para a criação de projetos e módulos devido ao seu grande conjunto de bibliotecas e suporta outras linguagens como o C, C++, PHP, XML etc (DEITEL, 2010).

3.2.3. Spring Security

O *Spring Security* é um *framework* que ajuda na proteção e na segurança de um projeto Java desenvolvido para web. O Spring realiza a autenticação e a interceptação de páginas e usuários dentro da aplicação, ou seja, limita o acesso a determinadas áreas e funções do sistema além de liberar o acesso à aplicação somente quando um usuário estiver logado no sistema (WEISSMANN, 2013).

3.2.4. JavaServer Faces

O JSF, ou como é mais conhecido *JavaServer Faces*, é uma tecnologia voltada para desenvolvimento web que utiliza modelos de interfaces gráficas baseado em eventos, definida pela *Java Community Process* o que torna padrão de desenvolvimento e facilita o trabalho dos desenvolvedores e valorizam os produtos, deixando-o com um visual mais atrativo e interessante (CORDEIRO, 2012).

3.2.5. MySQL

O MySQL foi criado na década de 1980 na Suécia; é um sistema gerenciador de banco de dados que se utiliza de uma linguagem de consulta estruturada, facilitando a consulta de dados pelos softwares (MILANI, 2006).

Uma das suas principais vantagens é a compatibilidade com diversos sistemas tanto pra web como para desktop, ele é gratuito, portanto a sua utilização é bem grande

no mundo inteiro. Esse banco de dados também é bastante utilizado devido a sua baixa exigência de hardware, facilitando assim a sua implementação em qualquer máquina com um recurso mínimo de hardware e exerce um grande desempenho e estabilidade na sua execução e manutenção dos dados. Outro ponto positivo também é a sua interface gráfica que permite ao desenvolvedor escolher interagir com o BD tanto pela linha de comando como pela interface gráfica (DEITEL, 2010).

Atualmente o MySQL pertence a companhia Oracle, e é um dos softwares gerenciados de banco de dados mais utilizado no mundo e por grande multinacionais e que destaca mais ainda a sua eficiência (CODEIRO, 2012).

4. ANÁLISE DE REQUISITOS

A análise foi feita com base em softwares semelhantes como *DropBox*, *4shared*, *FileUpload* dentre outros que apresentam a mesma funcionalidade, porém, será empregado um método de desenvolvimento mais sofisticado. Dentre estes requisitos estão os funcionais e os não funcionais.

4.1. REQUISITOS FUNCIONAIS

Nº	Descrição
01	Cadastrar Usuário
02	Cadastrar Administrador
03	Fazer Login
04	Fazer Upload de Arquivos
05	Gerenciar arquivos
06	Gerenciar usuários

4.2. REQUISITOS NÃO FUNCIONAIS

4.2.1. Requisito de Segurança

O sistema deve oferecer uma regra de segurança onde irá delimitar acessos a determinadas funções do sistema ao usuário, dando permissão total apenas ao administrador do sistema, onde para fazer login terá que informar um usuário e uma senha, e a partir deste usuário o sistema saberá o que deve restringir.

4.2.2. Requisito de Desempenho

O desempenho do sistema deve ser rápido e eficiente, sem erros. O sistema não pode travar ou perder quaisquer dados e arquivos que os usuários possam armazenar.

4.2.3. Requisito de Compatibilidade

Para ter acesso em qualquer computador ou dispositivo, independente do sistema operacional ou parte de hardware, o sistema deve ser executado perfeitamente conforme os requisitos definidos pelo *framework*.

4.2.4. Requisito de Implementação

Por se tratar de um sistema feito para a plataforma web, o computador terá que contar com um browser, o plugin do JRE (Java) e internet para acesso ao sistema.

4.3. ORÇAMENTO DO PROJETO

- Equipamentos

01 computador

Valor unitário: R\$ 1.000,00

Dias de (uso): 60 dias (considerar 400h, 4h de trabalho/dia, 02 dias/semana).

Depreciação: R\$ 1.000,00 / 12 meses (01 ano) = R\$ 83,33/mês.

Custo dia: R\$ 83,33 / 30 (qtde. de dias em 1 mês) = R\$ 2,78/dia.

Custo do computador: R\$ 2,78 * 60 (total de dias) = R\$ 166,80

Custo do computador = R\$ 166,80

- Softwares

NetBeans IDE 7.2.1 = sem custo.

Framework JSF = sem custo.

Framework Spring Security = sem custo.

Servidor Apache Tomcat = sem custo.

Banco de Dados MySQL = sem custo.

Custo total do projeto = R\$ 166,80.

4.4. CRONOGRAMA DO PROJETO

O Cronograma reflete o período de desenvolvimento do projeto.

	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
TAREFAS										
Levantamento Bibliográfico	X	X								
Levantamento de Requisitos		X	X	X						
Caso de Uso			X	X	X					
Programação					X	X	X	X	X	X
Testes								X	X	X
Escrita do Relatório										X

Figura 8 – Cronograma de Desenvolvimento

4.5. DIAGRAMAS

Este capítulo representa as unidades funcionais dentro do sistema, descrevendo os requisitos do sistema.

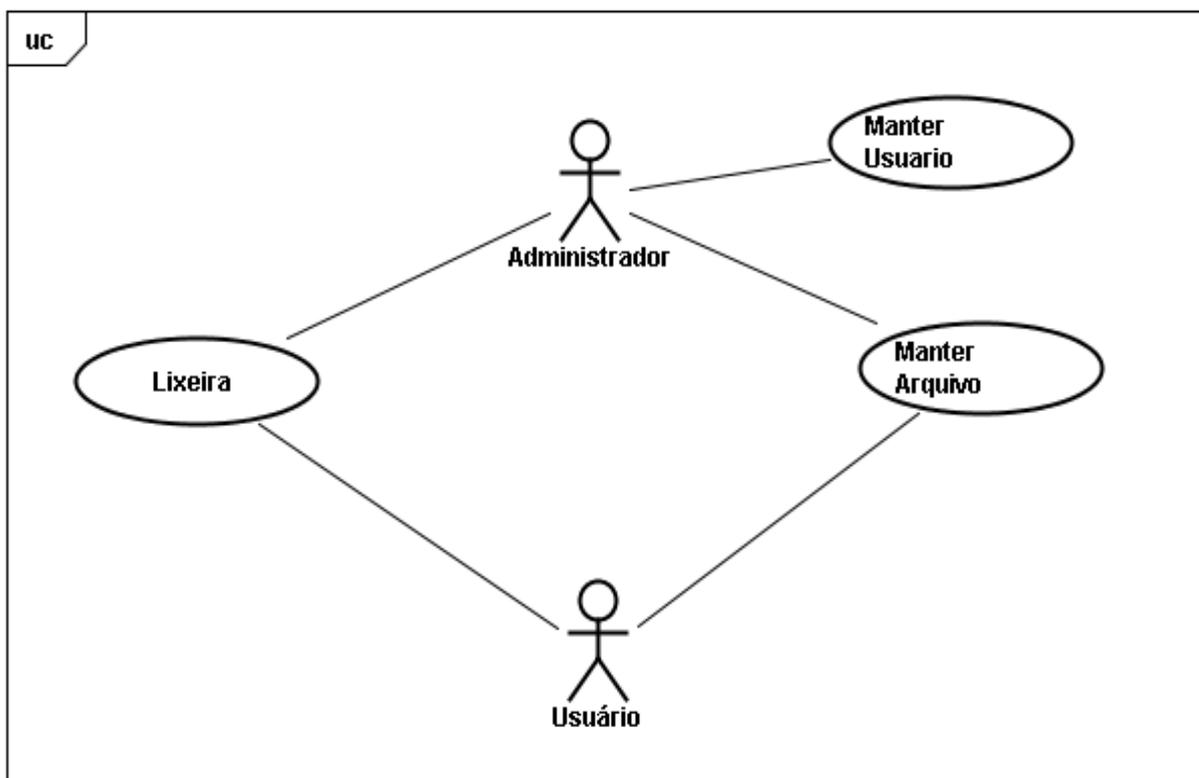


Figura 9 – Diagrama de Caso de Uso

4.5.1. DIAGRAMA DE CASO DE USO MANTER ARQUIVO



Figura 10 – Caso de Uso Manter Arquivo

Finalidade: permite ao usuário realizar o upload de arquivos.

Ator: Usuário

Pré-condição: Estar autenticado no sistema.

Evento Inicial: O usuário seleciona a opção de 'upload de arquivo'.

Fluxo principal:

- a. Seleciona a opção upload.
- b. O usuário confirma o upload do arquivo. **(A1)**
- c. O usuário escolhe a opção de upload. **(A1) (E1)**

Fluxo Alternativo: A1 – Cancelar o Upload

- a. O usuário cancela o upload.
- b. O sistema retorna ao passo A do fluxo principal.

Fluxo de Exceção: E1 – Arquivo muito grande

- a. O sistema informa que o arquivo é muito grande.
- b. O sistema retorna ao passo A do fluxo principal.

Pós-Condições: Terá um arquivo armazenado no sistema.

4.5.2. DIAGRAMA DE SEQUÊNCIA MANTER ARQUIVO

O diagrama de lixeira mostra a parte de exclusão e restauração dos arquivos da lixeira, isso permite uma confiabilidade para o usuário pois antes de excluir o arquivo completamente do sistema ele o envia para uma área que pode ser restaurado ou excluído definitivamente.

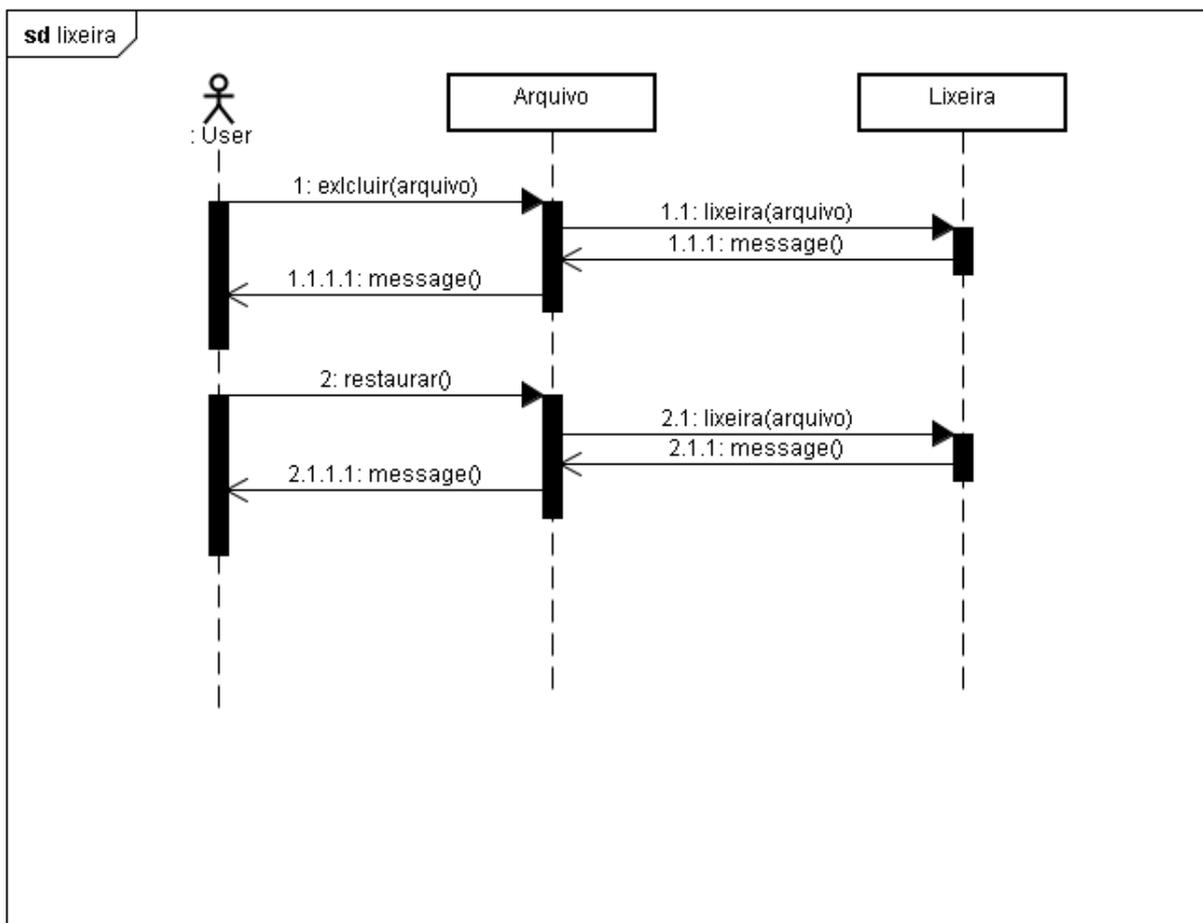


Figura 11 – Diagrama de sequencia – Manter Arquivo

4.5.3. DIAGRAMA DE CASO DE USO MANTER USUÁRIO



Figura 12 – Caso de Uso Manter Usuário

Finalidade: permite ao administrador consultar, alterar, incluir e excluir cadastro de usuários no sistema.

Ator: Administrador

Pré-condição: Estar autenticado no sistema.

Evento Inicial: O usuário seleciona a opção de 'manter usuários'.

Fluxo principal:

- Seleciona a opção manter usuários.
- O sistema oferece a opção de consulta de usuário. **(A1) (E1)**
- O sistema oferece a opção de alterar usuário. **(A1) (E1)**
- O sistema oferece a opção de incluir usuário. **(A1) (E2)**
- O sistema oferece a opção de excluir usuário. **(A1) (E3)**
- Encerra o recurso.

Fluxo Alternativo: A1 – Cancelar

- a. O Administrador cancela a operação.
- b. O sistema retorna ao passo A do fluxo principal.

Fluxo de Exceção: E1 – Usuário não cadastrado

- a. O sistema realiza a busca e verifica que o usuário não possui cadastro.
- b. O sistema oferece a opção de cadastrar usuário. **(A1)**
- c. O sistema retorna ao passo A do fluxo principal.

E2 – Usuário já possui cadastro

- a. O sistema realiza a busca e verifica que o login e/ou nome do usuário já consta no sistema.
- b. O sistema oferece a opção de consulta no sistema. **(A1)**
- c. O sistema retorna ao passo A do fluxo principal.

E3 – Usuário possui arquivos

- a. O sistema realiza uma busca e verifica que o usuário possui arquivos no sistema.
- b. O sistema oferece a opção de excluir os arquivos. **(A1)**
- c. O sistema retorna ao passo A do fluxo principal.

Pós-Condições: A inclusão, exclusão, consulta ou alteração de usuários no sistema foi realizada, deixando o sistema assim melhor adequado as condições requisitadas pelo administrador.

Caso de teste: Ao tentar excluir um usuário que possui arquivos o sistema retorna os arquivos armazenados pelo usuário.

4.5.4. DIAGRAMA DE SEQUÊNCIA MANTER USUÁRIO

O diagrama de manter usuário mostra a manutenção que o administrador pode fazer em relação ao cadastro de usuários do sistema, podendo editar, excluir e cadastrar os usuários. Nesse diagrama somente o usuário que é administrador do sistema pode ter acesso a essa opção.

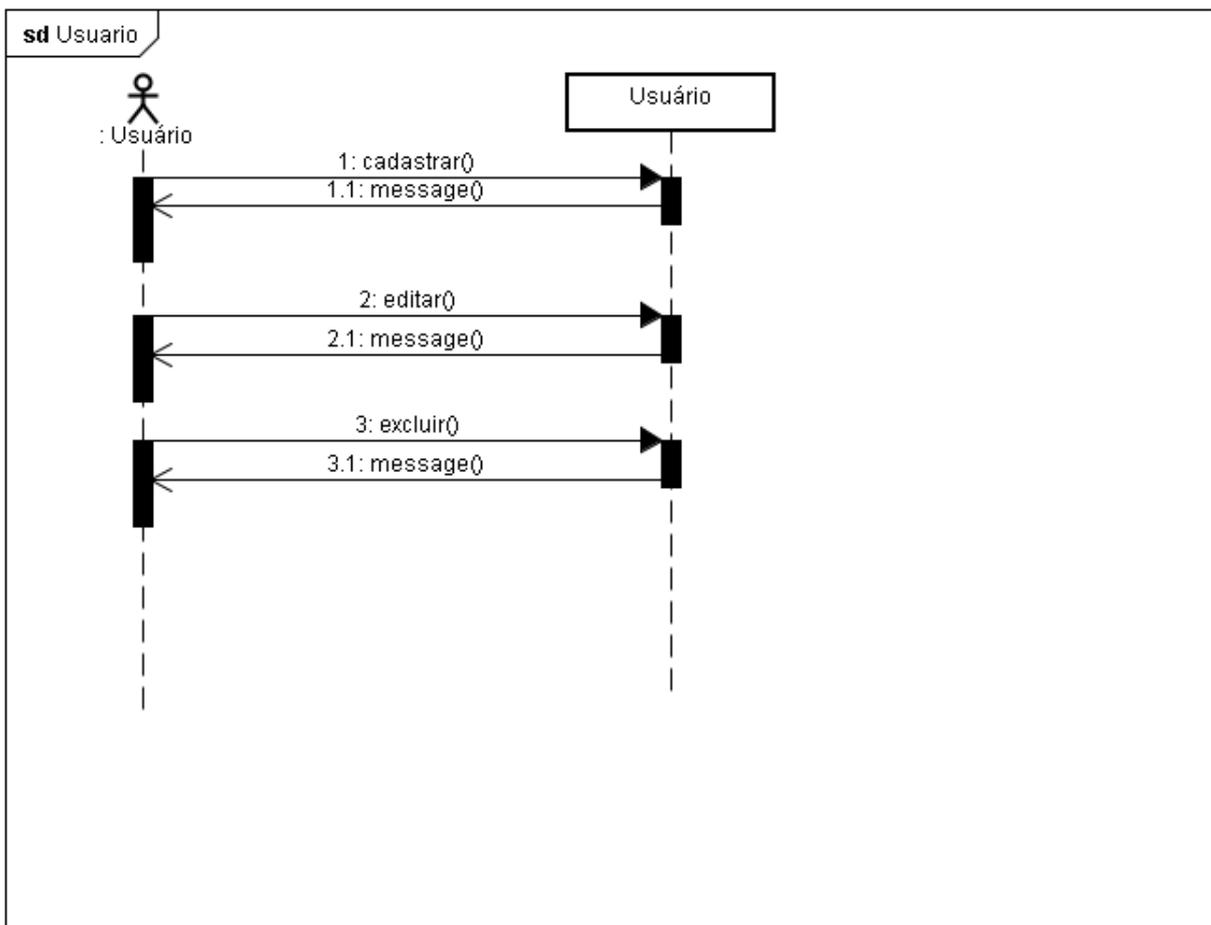


Figura 13 – Diagrama de sequencia – Manter usuário

4.5.5. DIAGRAMA DE CASO DE USO



Figura 14 – Caso de Uso Manter Arquivo - Administrador

Finalidade: permite ao administrador realizar o upload e download de arquivos.

Ator: Administrador

Pré-condição: Estar autenticado no sistema.

Evento Inicial: O administrador seleciona a opção de 'upload de arquivo'.

Fluxo principal:

- a. Seleciona a opção upload.
- b. O administrador confirma o upload do arquivo. **(A1)**
- c. O administrador escolhe a opção de upload. **(A1) (E1)**

Fluxo Alternativo: A1 – Cancelar o Upload

- c. O administrador cancela o upload.
- d. O sistema retorna ao passo A do fluxo principal.

Fluxo de Exceção: E1 – Arquivo muito grande

- c. O sistema informa que o arquivo é muito grande.
- d. O sistema retorna ao passo A do fluxo principal.

Pós-Condições: Terá um arquivo armazenado no sistema.

Caso de teste: Ao digitar login ou senha incorretos mostrar uma mensagem de erro, e o sistema retorna a página de login.

4.5.6. DIAGRAMA DE SEQUÊNCIA UPLOAD ARQUIVO

Os diagramas a seguir mostram as principais atividades realizadas dentro do sistema pelos usuários do sistema, assim como manutenção de usuários e arquivos e a realização do upload e download dos mesmos.

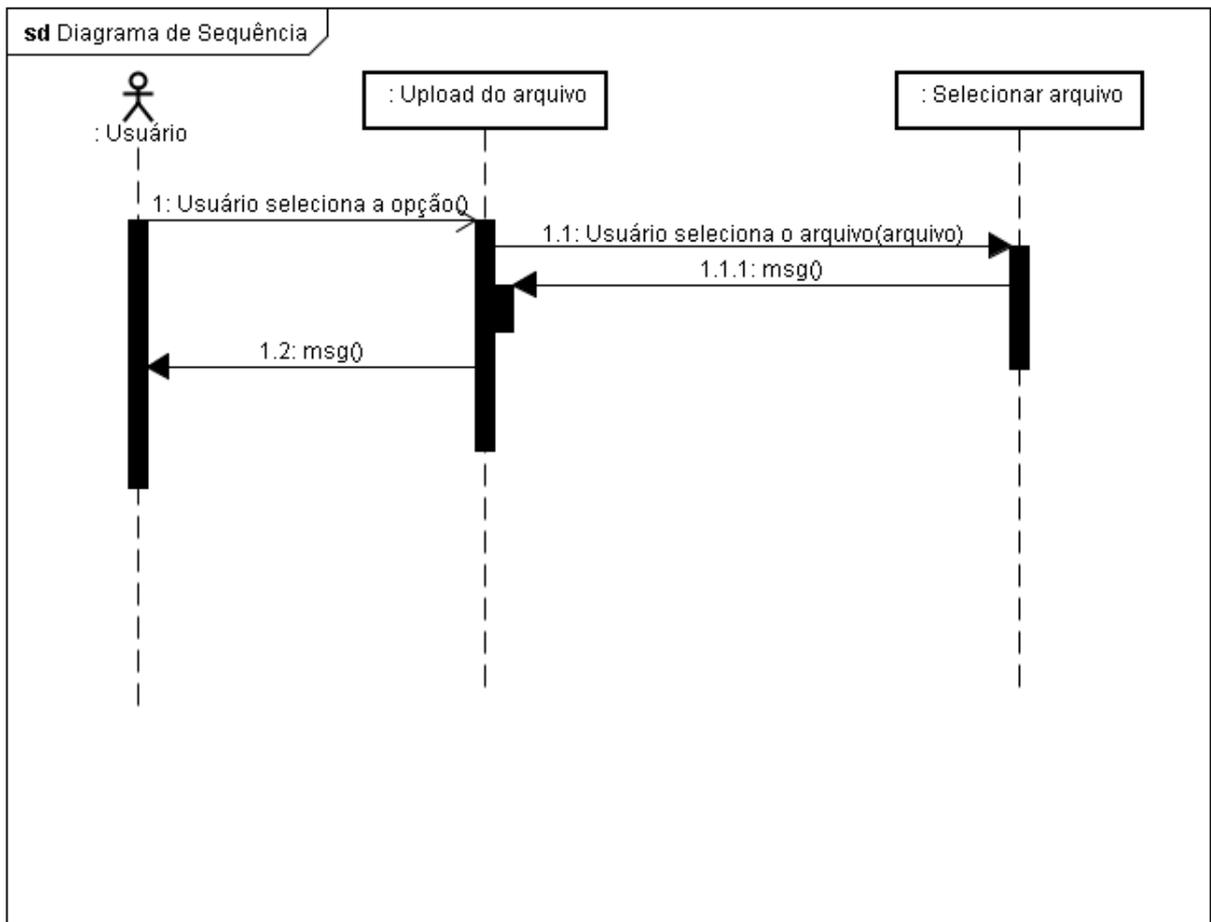


Figura 15 – Diagrama de sequencia – Upload de arquivo

4.5.7. DIAGRAMA DE SEQUÊNCIA DOWNLOAD ARQUIVO

O diagrama de sequencia de download de arquivos mostra o caminho percorrido pelo usuário dentro do sistema para fazer esta ação.

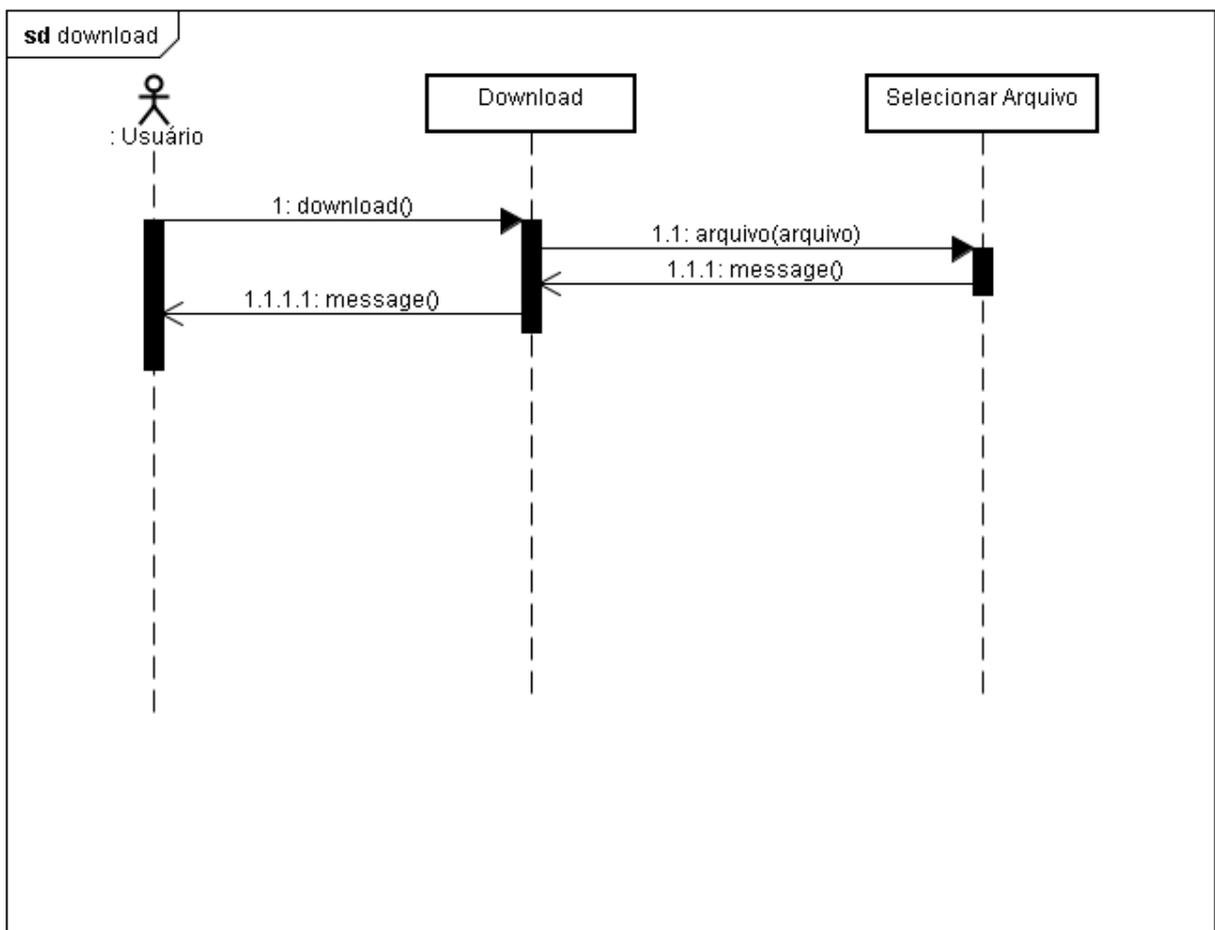


Figura 16 – Diagrama de sequencia – Download de arquivo

4.5.8. DIAGRAMA DE CLASSE

O diagrama de classes representa a estrutura do sistema, recorrendo ao conceito de classe e suas relações. O modelo de classes resulta de um processo de abstração onde são identificados os objetos relevantes do sistema em estudo.

Este diagrama representa a estrutura e relação entre as classes do sistema desenvolvido.

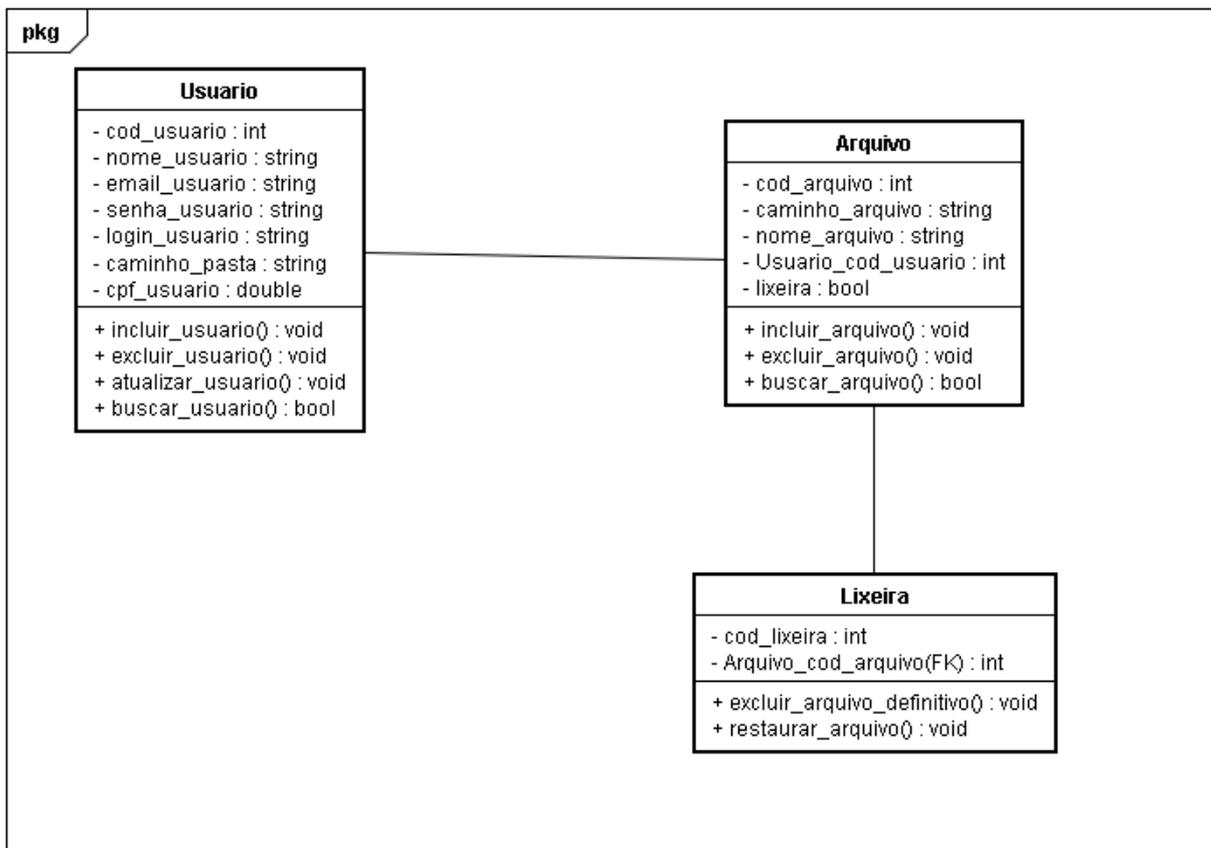


Figura 17 – Diagrama de Classes

4.5.9. MODELO DE ENTIDADE-RELACIONAMENTO

O Modelo Entidade-Relacionamento tem o objetivo de representar as estruturas de dados da forma mais próxima do mundo real dos negócios.

Este diagrama representa o modelo conceitual de negócio do banco de dados.

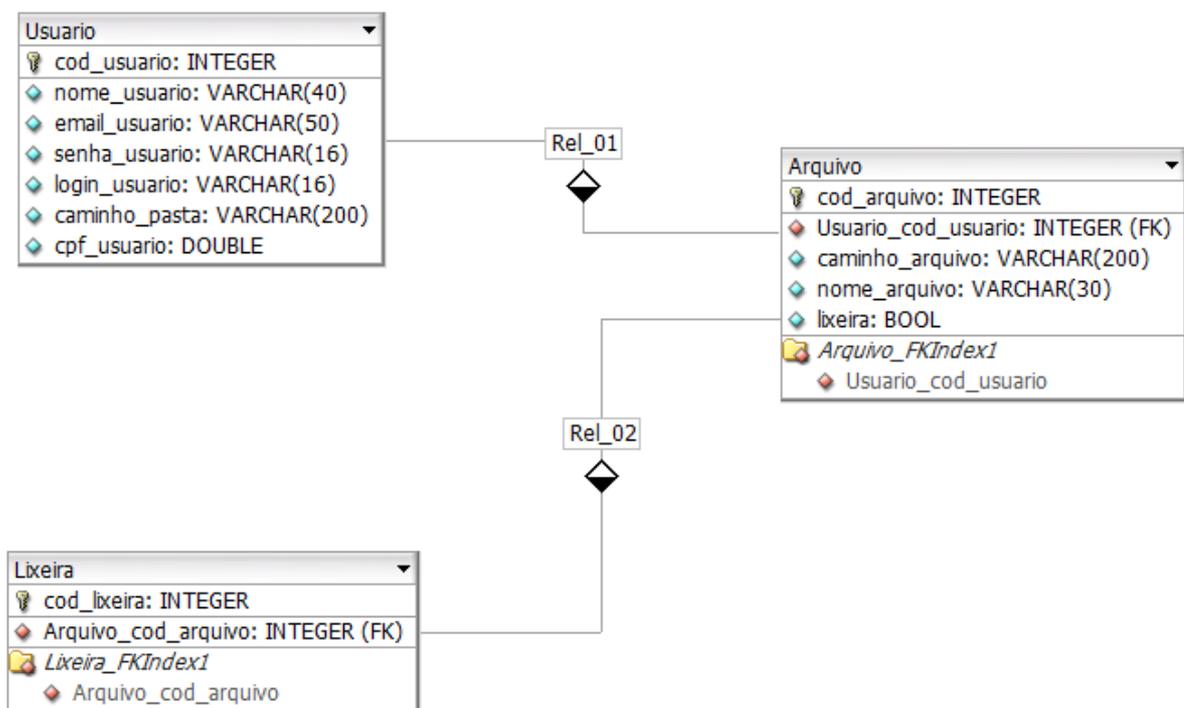


Figura 18 – Modelo de Entidade-Relacionamento

4.5.10. PROCESSOS PRINCIPAIS

Os diagramas do processo principal mostram detalhadamente o upload e download de arquivos dentro do sistema.

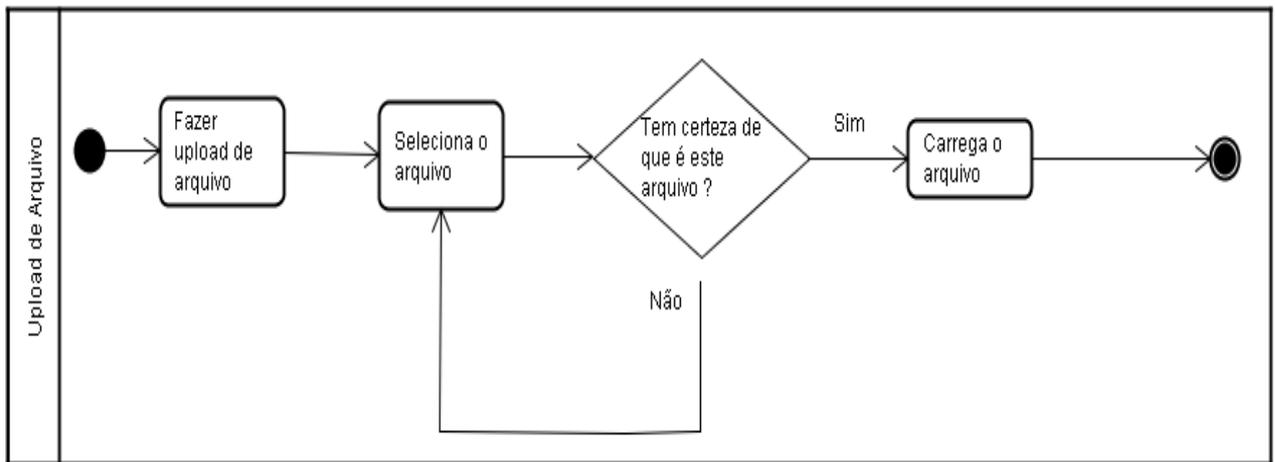


Figura 19 – Processo de Upload

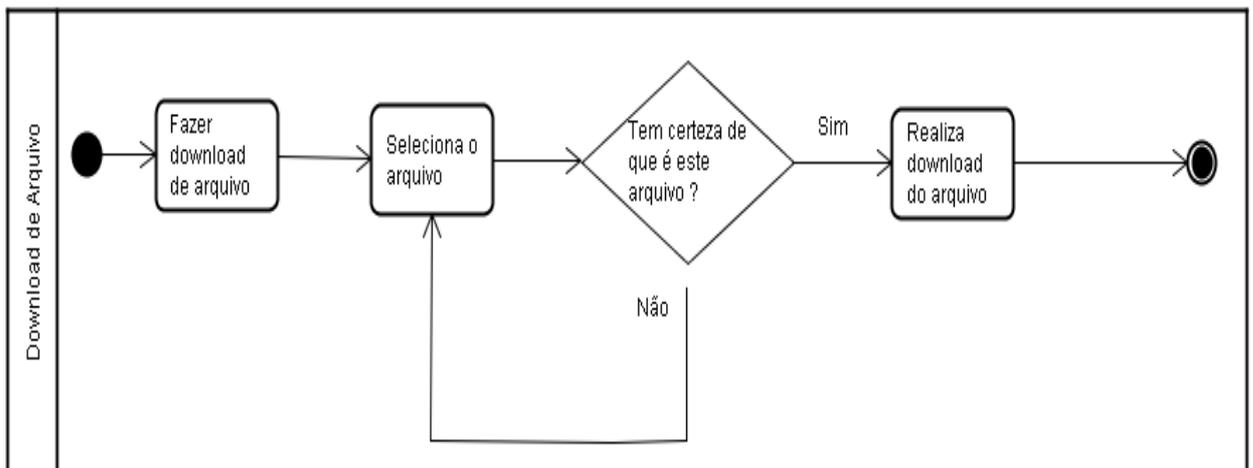


Figura 20 – Processo de Download

5. CONCLUSÃO

O projeto teve como foco principal mostrar de forma prática a implementação de uma ferramenta de segurança de um sistema web. A segurança traz comodidade para o usuário, isso faz com que a confiabilidade vinda de um possível cliente seja mais satisfatória.

A experiência de desenvolver esse projeto foi indescritível, pois o conceito adquirido foi mais importante do que até mesmo o próprio desenvolvimento do software, pondo a prova o conhecimento adquirido na faculdade, isso incentivou ainda mais a buscar novas soluções fora da instituição de ensino e compartilhando experiências com amigos do mesmo ramo de atividade.

Esse software traz comodidade ao usuário, pois armazenar os seus arquivos na web, facilita e traz segurança para trabalhadores, um exemplo que podemos citar é um consultor, pois armazenando seus arquivos na web ele pode viajar tranquilamente sem a preocupação ou o incomodo de ficar se preocupando com notebook ou ao mesmo com pendrive, já armazenando os arquivos no software ele sabe que onde ele estiver e com um sinal de internet ele consegue ter em mãos os arquivos.

Portanto, conclui-se que o software foi implementado de acordo com a necessidade de uma empresa, pode trazer benefícios excelentes, melhorando a viabilidade de uso dos arquivos por parte dos usuários, lembrando que deixa os arquivos mais resistentes a vírus, impedindo que haja perda de informação. Podemos dizer então que o software futuramente pode ser uma grande inovação, mudança ou expansão no conceito de intranet, por exemplo.

REFERÊNCIAS

GOMES, Yuri. **Java na Web com JSF, Spring, Hibernate e NetBeans6**. Editora Ciência Moderna, 2008.

CORDEIRO, Gilliard. **Aplicações Java para web com JSF e JPA**. Casa do Código, 2012.

WEISSMANN, Henrique. **Vire o jogo com Spring Framework**. Casa do Código, 2012.

DEITEL, Paul; DEITEL, Harvey. **Java como programar**. 8ª edição. Pearson, 2010.

JOHNSON, Rod. **Expert One-on-One J2EE and Development**, Wrox, 2002.

MILANI, André. **MySQL – Guia do Programador**, Novatec, 2006.

VUKOTIC, Aleksa; GOODWILL, James. **Apache TOMCAT 7**, Apress, 2011.

Spring Framework Reference Documentation. Disponível em <http://docs.spring.io/spring/docs/4.0.0.BUILD-SNAPSHOT/spring-framework-reference/htmlsingle/>. Acessado em: 03 ago, 2013.